STORAGE MANAGEMENT

(1) Other than the CPU, main memory is an important resource of a computer system which must be properly managed for overall system management.

(2) The memory management of an operating system takes care of its requirement.

(3) Its job is to keep track of which part of memory are in use and which part are not in use.

(4) To allocate memory to processes which they need it and deallocate them they are done.

LOGICAL ADDRESS (VIRTUAL ADDRESS)

The address generated by the CPU is known as logical or virtual address.

OR

The address which are used in the program are called logical address.

PHYSICAL ADDRESS

The real or physical is the actual main memory provided in the systems, it is directly addressed by CPU called physical address

OR

Any location in the main memory is uniquely identified by an address. This is called physical address of main memory or RAM.

GICAL ADDRESS SPACE

The set of all logical addresses is called logical address space.

PHYSICAL ADDRESS SPACE

The set of all physical addresses is called physical adress space.

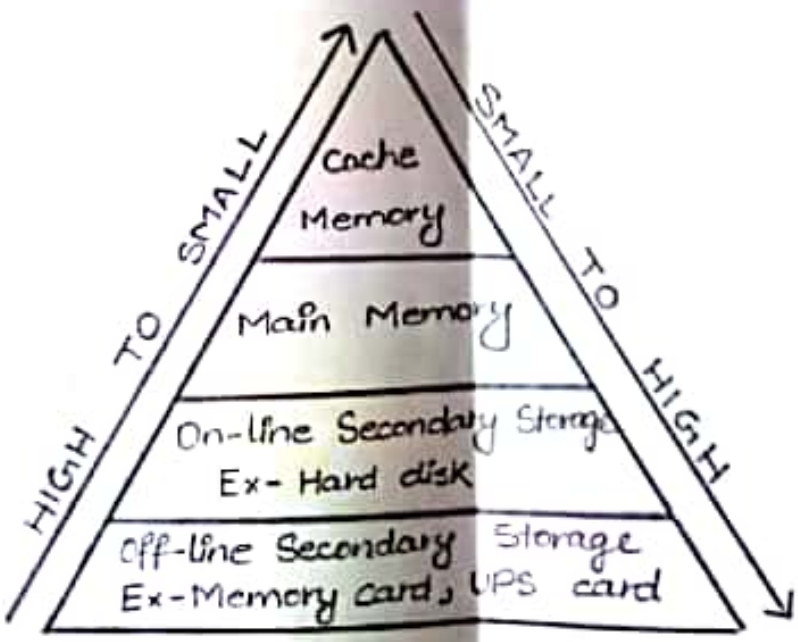
MEMORY MAPPING

The procedure for conversion of logical to physical address is called memory mapping.

MMU (MEMORY MANAGEMENT UNIT)

The hardware unit that convert a logical address to physical adress is called MMU (Memory Management Unit).

STORAGE HIERARCHY



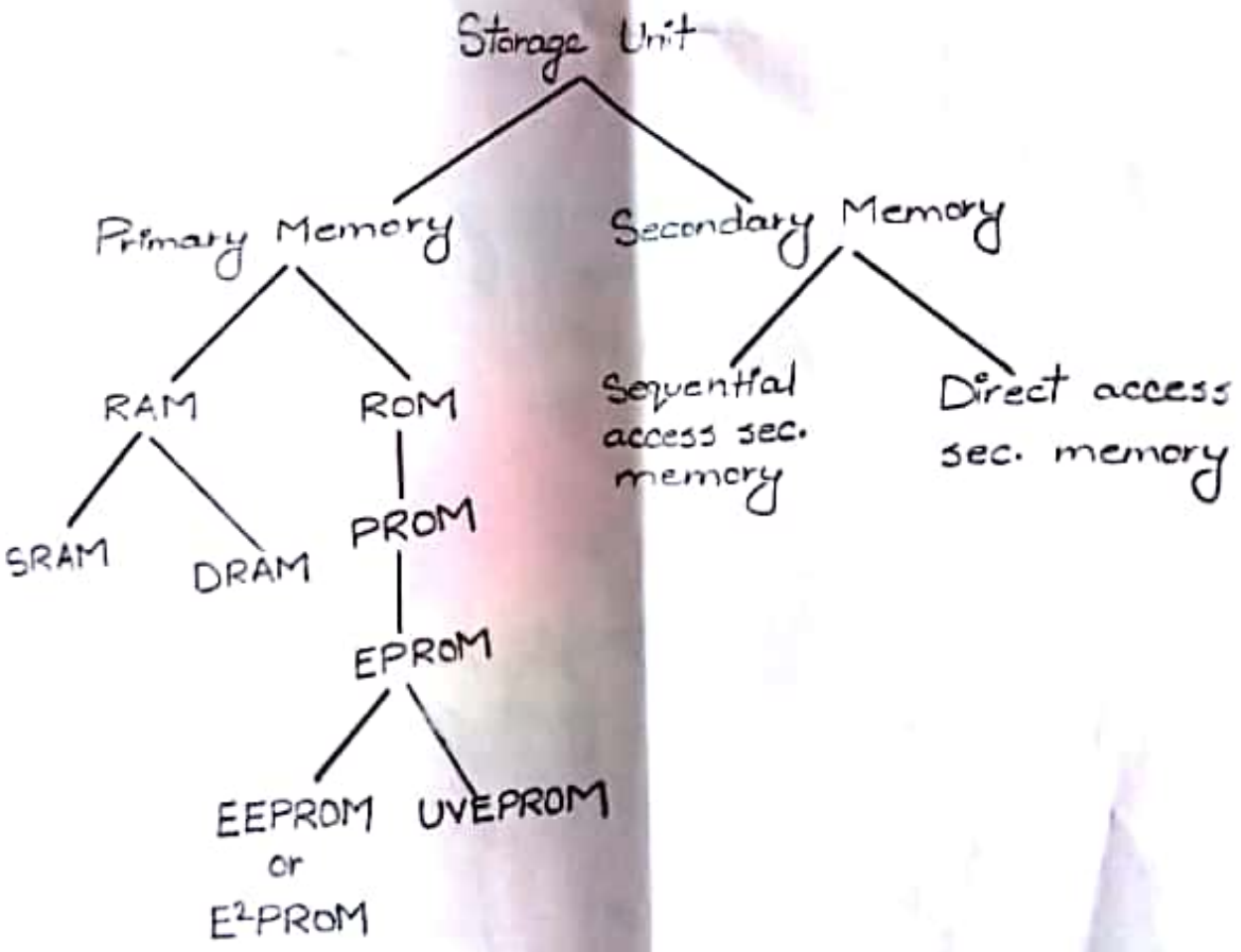
It includes cache memory, main memory and secondary storage.

2) As we move up the pyramid, we encounter storage element which has faster access time, higher cost per bit store and smaller capacity.

3) As we move down the pyramid, we encounter storage element, which have slower access time, lower cost per bit store and larger capacity.

4) Cache Memory → Main Memory → On-line Secondary storage → Off-line Secondary storage.

STORAGE ORGANISATION



STORAGE UNIT

It is used to store the data.

It is also the part of hardware.

A computer uses two types of storage:-

a) Main Memory, and

b) Secondary Memory

a) Main Memory

Main Memory temporarily stored instruction or data to be executed by the computer. The Primary storage hold information only when the computer system is on. As soon as the computer system switched off or reset, the information held in the primary storage is disappeared.

b) Secondary Memory

It is used to take care the limitation of primary memory. It is the permanent storage device which retain(reserve) information even when the computer system is switched off or reset.

Primary Memory is of two types

I. RAM (Random Access Memory)

II. ROM (Read Only Memory)

I. RAM

RAM is temporarily r/w memory of a computer system and faster than secondary memory.

ROM

Programme held in ROM are called firmware. They are stored permanently and are read for use when the computer is switched on. Users cannot write into a ROM. Its contents are written at manufacturing time i.e. POST (Power on Self Testing). ROM store fonctions such as square root, cosine, exp, log, sine etc.

PROM (Programmable Read Only Memory)

Programme held in ROM are called PROM.

EPROM (Erasable Programmable Read Only Memory)

Once a information stored in ROM or PROM, chip cannot be altered or erased. However, another type of erasable programmable read only memory is developed which erase the stored information and restore the new information or programme.

There are two types of EPROM:-

i) EEPROM or E^2PROM

ii) UVEPROM

i) EEPROM (Electrically Erasable Programmable Read Only Memory)

Erased with the help of high voltage electricity.

ii) UVEPROM (Ultra-violet Erasable Programmable Read Only Memory)

Erased by ultra-violet light.

RAM is of two types:
I. SRAM (Static RAM)
II. DRAM (Dynamic RAM)

Static RAM	Dynamic RAM
(i) Static RAMs retain stored information only as long as the power supply is on.	(i) Dynamic RAM loses its stored information in a very few seconds (nanoseconds) even though the power supply is on.
(ii) It has very high speed.	(ii) It has low speed.
(iii) It consume more power.	(iii) It consume less power.
(iv) It is costlier	(iv) It is cheaper

b) Secondary Memory

It is of two types:

I. Sequential access secondary memory.

II. Direct access secondary memory.

I. Sequential access Secondary memory:-

A sequential access storage device is one in which the arrival at the location desire may be proceeded sequentially through other location.

Ex- Magnetic Tape.

II. Direct access Secondary memory:-

A direct access storage device is one in which the arrival is literally available at random i.e. it is available in any order.

Ex- • Optical disk such as CD-ROM, DVD, WORM etc.

agnetic disk such as Floppy disk, Hard disk etc.
Removable disk such as pen-drive, memory card etc.

MEMORY ALLOCATION

(1) The procedure of assigning memory to a process is called memory allocation.

(2) There are two types of memory allocation:
- → Contiguous memory allocation
- → Non-contiguous memory allocation.

(3) In contiguous memory allocation, process is loaded in one or more contiguous blocks of memory. It is sufficient that contiguous memory allocation loaded in contiguous form. Its (process) are not loaded in scattered or fragmented order.

(4)

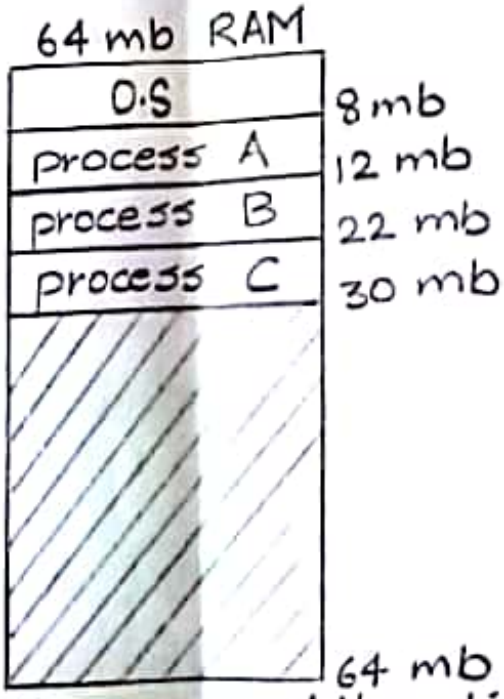


Fig. Contiguous Memory Allocation

In the above figure, main memory as 64 mb and the processes are loaded in contiguous form (not in scattered or fragmented order.)

In non-contiguous memory allocation, a process can be loaded in scattered way. In other word, a process may not be loaded in contiguous memory allocation.

(6)

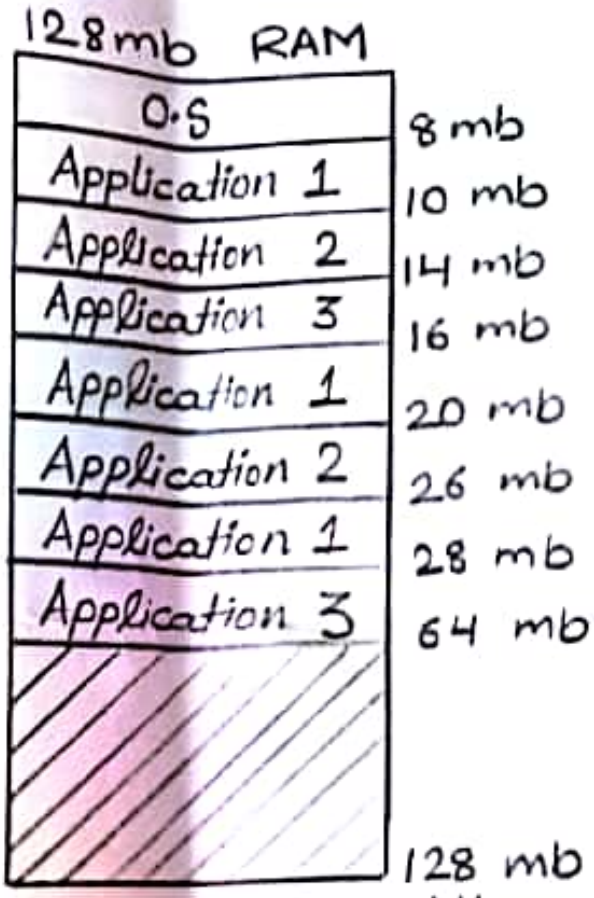


Fig. Non-contiguous Memory Allocation

In the above figure, main memory as 128 mb and the processes are loaded in scattered or fragmented order.

Single User Contiguous Storage Allocation

(7) The single user contiguous storage allocation is used in uniprogramming system (in which one job is processed by the system at a time) and multitasking system (in which more than one job is processed by the system at a time.)

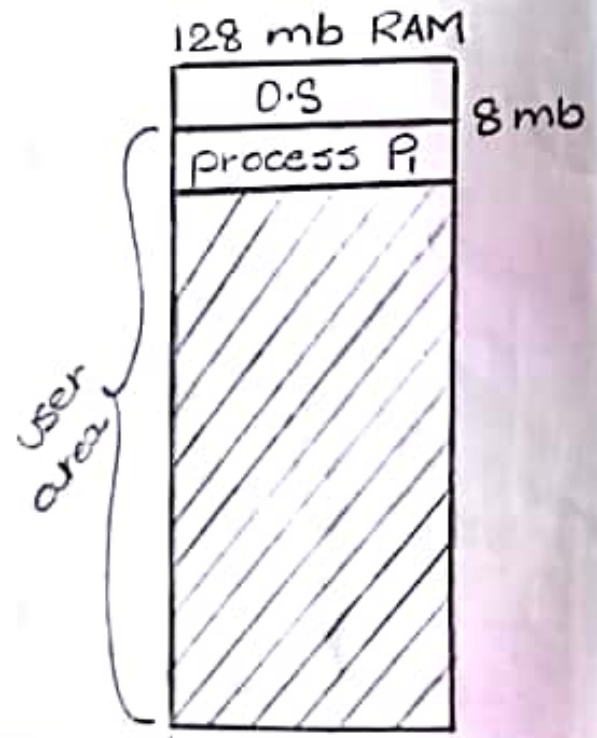
operating system load and all the system resource are exclusively available for the single user.

(3) In this memory management strategy, the operating system resides in one part of the memory and an entire remaining part of the memory is available for user. The currently active user process.

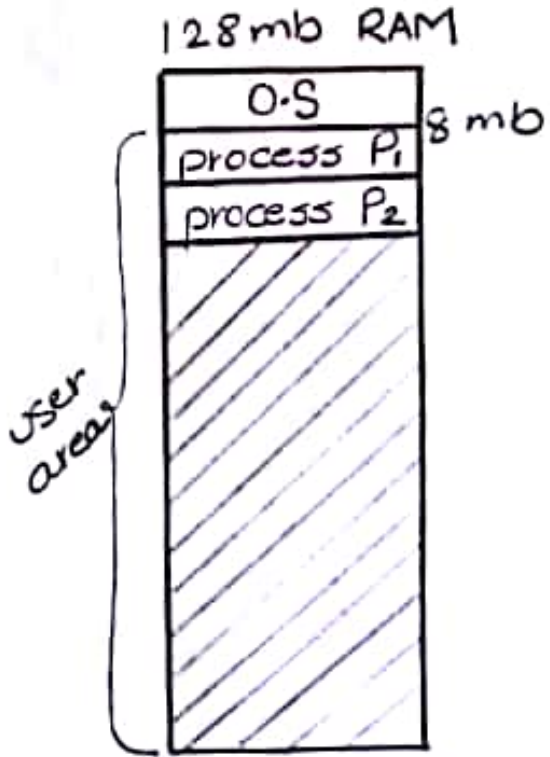
(4) In this scheme, operating system load the program to be executed from the disk into the user area of the memory and executes it.

(5) When the processes finish, the operating system cleans up the user area of the memory and load the next program to be executed.

(6) The two figure shows single user contiguous memory allocation



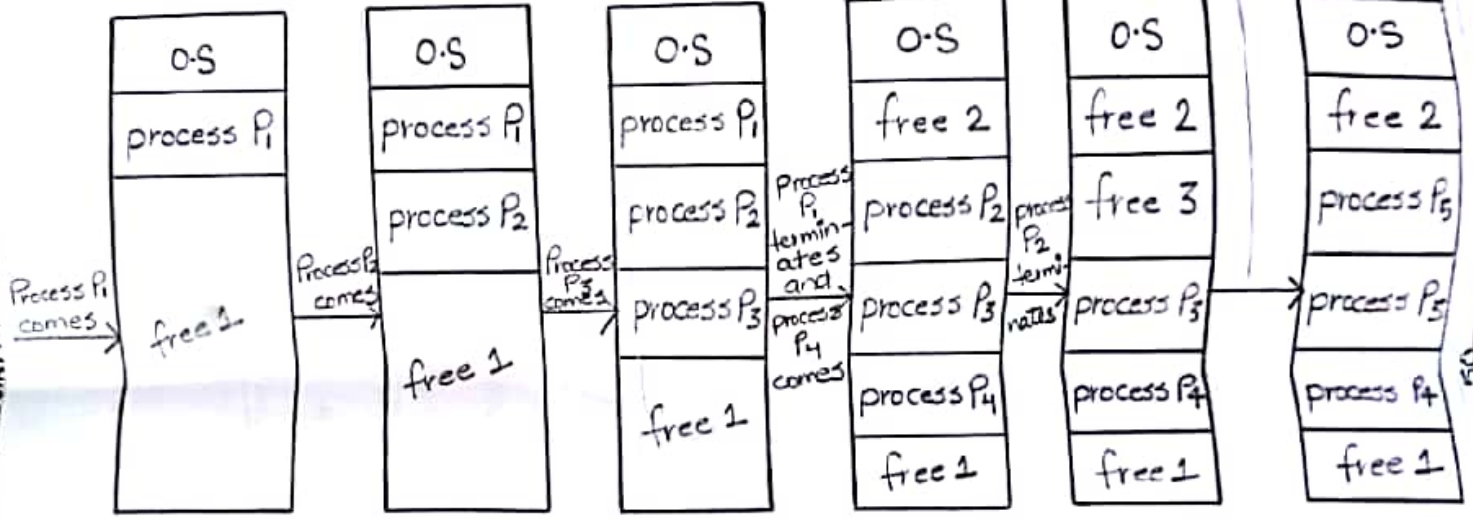
Single tasking/uniprogramming
fig.1



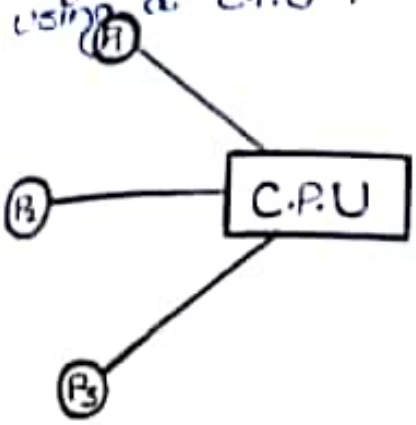
Multitasking
fig.2

Variable Portion Multiprogramming

process P_5 comes but not fit in free 1 then it goes to free 2 but not fit in free 2 then it goes to free 3 and fit in free 3.



Multiprogramming operating system allows multiple users to execute multiple programs using a C.P.U.



(2) It is the memory management scheme in which all memory is available for the user space.

(3) When a process arrives, only as much memory as needed by it is allocated to its keeping. The rest available are to satisfy future request.

(4) Operating System maintain a table track of which part of memory keep to use and free to be occupied.

(5) When a new process arrives, the operating system searches for a free block i.e. large enough for this process.

(6) If the free block is too large, it is split into two parts. One part is used for memory requirement of the process and is allocated to the process. The other part contain the remaining memory for the process.

When a process terminates, it releases the portion allocated to it.

UAL MEMORY

Conventional memory management scheme discussed until now, suffer from the following two main limitation. These are:

(i) A process cannot be loaded and has keep waiting for its executing to start until sufficient free memory for loading the entire process scheme become available. This may delay process's turnaround time.

(ii) A process cannot be loaded in a system whose memory delay a process's turnaround time.

(2) Three - basic concept for the realisation of virtual memory are:

(a) On-line Secondary storage.

(b) Swapping

(c) Demand Paging

(a) On-Line Secondary Storage ⇒ It is a secondary storage device i.e. hard disk whose capacity is much larger than the main memory and which is always kept on-line to the system. High speed disk storage is used for this purpose.

(b) Swapping ⇒ Swapping is the process of transferring a block of data from on-line secondary storage to the main memory and then main-memory to on-line secondary storage.

Transferring of data from on-line secondary to main memory is known as a swapping-in of data and transferring of data from main memory to on-line secondary is known as swap-out of data.

Demand Paging: In a virtual memory system, all the processes are partitions into pages (logical address space) and resides in on-line secondary storage. Pages are swapped-in and out of the main memory.

A program is initially stored in the on-line secondary storage. When a page in the program is required, it is swapped-in to the main memory. It is called demand paging because until a page is not required, it is not loaded.

(3) Based on the way, virtual memory is realised.

(4) Virtual memory is often described hierarchy of two storage system:

(i) On-line disk storage
(ii) Main memory (RAM)

The Operating system manages the two storage system in such a way that the users of the system feels that they have accessed a single, large, direct addressable and fast main memory.

(5) This is from the point of view of application program, virtual (realise/imagine/feeling) size of the available main memory is to be unlimited.

(6) Virtual memory is storing data and instruction of a program in the on-line secondary storage and when they required, they are loaded in the main memory.

In other word, the part of a program is required at any instant is loaded in the main memory while the rest of the program is in the secondary requirement.

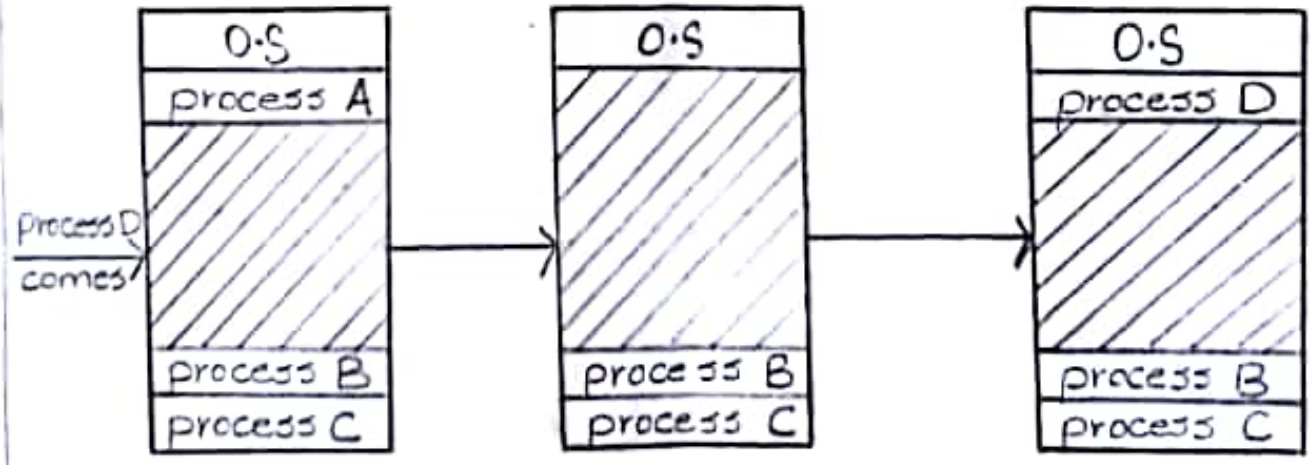
SWAPPING

(1) Swapping is the technique to remove process from the main memory and storing it in the on-line secondary memory or secondary memory usually hard-disk.

(2) Swapping should be very fast so that it doesnot become a large overhead to execute a process.

(3) A process is swapped-out to make room in the main memory for the processes.

(4) The area of the disk where the swap-out processes are stored is called as the swap area.



process A → swap-out
process D → swap-in

EXPLANATION

(i) In the above figure, the concept of swapping is the multitasking system. It occupy a part of the memory. Various user processes use rest of the memory known as user program area of the memory.

(ii) Initially, memory is allocated to process A, B, C.

(iii) process 'D' is required to be loaded in the memory for execution.

(iv) The available space is not sufficient for the process 'D'.

(v) process 'A' is swapped-out of the memory and is stored in on-line secondary storage or secondary memory (i.e. hard disk.)

(vi) To create space for process 'D', process 'A' is loaded into on-line secondary storage.

PAGING

(1) Paging is a method of non-contiguous memory allocatio.

(2) In paging, logical address space is divided into fixed-size blocks, known as "pages".

(3) Physical address space is also divided into fixed-size blocks, known as "frame."

(4) A page is mapped into a frame.

(5) Individual pages and frames are recognized by unique number known as page number and frame number respectively.

(6) Size of a page is equals to (=) the size of a frame.

(7) Page Number denotes unique identification. Rest of logical address denotes Offset.

(8) Frame Number denotes unique identification. Rest of physical address denotes Offset.

(9) Example-

Logical Address (page) of 13 bit.

Page no.	Offset				
011	10	10	10	10	10

The higher 13-10 i.e. 3 bit denotes the page number and rest of the 10 bit denotes Offset.

Frame no.	Offset				
011	10	10	10	10	10

frame	00
frame	01
frame	10
frame	11

The logical address 011 1010101010 denotes the page number 011, the higher order left 3 bit.

<10> Operating system maintain a table to convert a logical address into its corresponding physical address. This table is called MPT (Memory Page Table:)

page no.	frame no.	presence bit
000	00	0
001	01	1
010	10	0
011	11	1

Fig. Memory Page Table

<11> There are two fields in the page table. The first field stored the corresponding frame number and the next field is called presence bit which shows page is available or not.

Ex- page 6 is stored in the 13th frame.

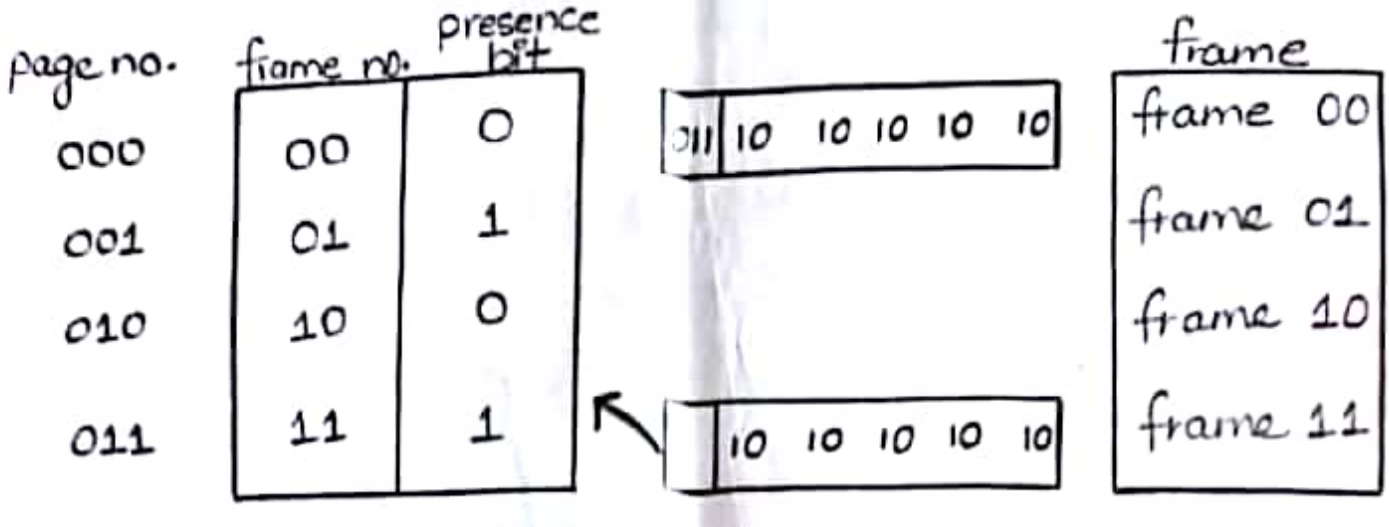


Fig. Paging system

PAGE FAULT

(1) In demand paging, M·M·U (Memory Management Unit) also maintain an index table for address translation (Logical to physical and physical to logical) known as "Page Table". The page Table stores the information in the main memory.

(2) When a page is required, the operating system scan the page table to find out the page. If the page is found, operating system continue processing. In case the page is not present in the main memory, page fault occurs.

(3) A page fault is not a fault or errors rather it indicates a situation that the page requested by a program is not currently available in the main memory.

When a page fault occurs, M·M·U swap-in the requested page from the in-line secondary memory into the main memory.

In case there is no free space in the main memory, MMU finds free space if not available. MMU swap-out an unused page from the main memory and stored unused page in the on-line secondary memory for creating space in the main memory. Then the required page is loaded in the main memory and finally page table is updated.

Page Replacement Algorithms

A page replacement algorithm is the logic or policy regarding how to select a page to be swapped-out-from the main memory to create space for the page known as requested page, which caused a page fault.

There are several page replacement algorithm. These are as follows:

(1) First-in-First-out (FIFO) Replacement Algorithm.
(2) Optimal page Replacement Algorithm.
(3) Least Recently Used (LRU) Replacement Algorithm.

(1) First-in-First-out (FIFO) Replacement Algorithm

The First-in-First-out Replacement Algorithm is the simplest of all the page replacement algorithm. It conveys a basic idea that when a page fault occurs, the oldest page in the main memory is to be swapped-out of the main memory to create room for the required page that needs to be executed.

For Example: Page number 14, 21 and 18 are present in the memory and page number 21 was loaded first followed by 14 and 18.

If the page number 30 is required to be accessed by the C.P.U, it results in Page fault because page number 30 is not available in the main memory.

The page loaded first ie. page number 21 is swapped-out of the main memory and stored in the on-line secondary memory to create room for the page number 30. Then 30 is loaded in the main memory.

(2) Optimal page Replacement Algorithm (will not been)

(1) Optimal page Replacement Algorithm is considered as the best possible page replacement algorithm.

(2) In virtual memory, theoretically it is difficult to implement.

(3) According to the page replacement policy, the page in the main memory which is not been referred or requested for the longest time is swapped-out from the main memory and creates room for the requested page.

For Example: page 14, 21 and 18 are present in the main memory. page 82 is required to be loaded resulting in a page fault since page 82 is not in the main memory at present.

page 14 is not required till 2000 instruction. page 21 is not required till 1000 instruction and, page 18 is not required till 1500 instruction.

This means that page 14 is one that will not been accessed by the CPU for the largest time. Page 14 is swapped-out from the main memory and creates room for the page 82.

(3) Least Recently Used (LRU) Replacement Algorithm (Has not Been)

(1) LRU algorithm uses information about the pages accessed in recent past to predict the near-future.

(2) LRU algorithm states that when a page fault occurs, the page that has not been referred to for the largest time is swapped-out of the main memory to create space for the requested page that has caused the page fault.

(3) Implementation of LRU can be done in various ways. One of the common method is to apply the LRU scheme for virtual memory management which is done by using an array.

The Array stores the information about the page present in the main memory.

The front end of the array stores the page accessed recently.

The rear end (back end) of the array stores the pages that has not been accessed for the longest period of time.

However, a page that is present in the main memory is accessed. The information about the page is shifted to the front end of the array.

If a page fault occurs, the

page indicated by the rear end of the array is swapped-out of the main memory and requested page is swapped-in the main memory.

Information about the page which is swapped-in is stored in the front end of the array.

For Example→ The array stores the following four page number

12, 56, 27, 61.

page 12 is in the front end of the array. This indicates that page 12 has been accessed recently.

page 61 is at the rear end of the array. This indicates that page 61 has not been referred for the longest time.

page 27 is accessed. No page fault occurs because page 27 is present in the main memory. Information about the page 27 is shifted to the front end of the array. The array becomes

{27, 12, 56, 61}

page 42 is referred to access. A page fault occurs, since page 42 is not in the main memory.

page 61 is at the rear end of the array which indicates that it has not been accessed for the longest time. So, page 61 is swapped-out from the main memory. The array becomes

{42, 27, 12, 56}

SEGMENTATION

(1) Memory segmentation is a technique of non-contiguous memory allocation.

(2) In segmentation, logical address is divided into certain unequal size of chunks known as segment.

(3) A logical address consists of two parts
   i) Segment number
   ii) Offset

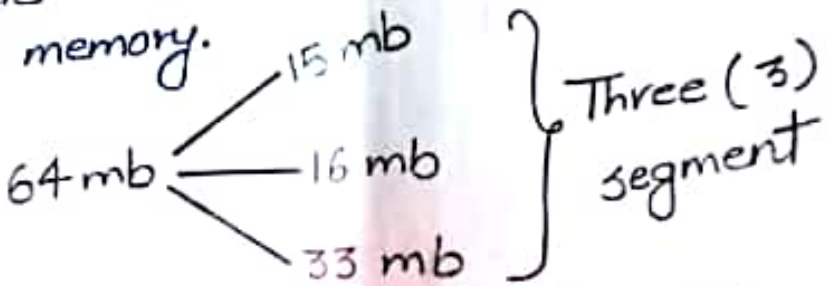
(4) Example- logical address

Segment no.	Offset

3 : 1000

3:1000 indicate that 1000 offset and segment is 3.

(5) An application can consist of three modules. Each module is loaded in distinct segment in the main memory.



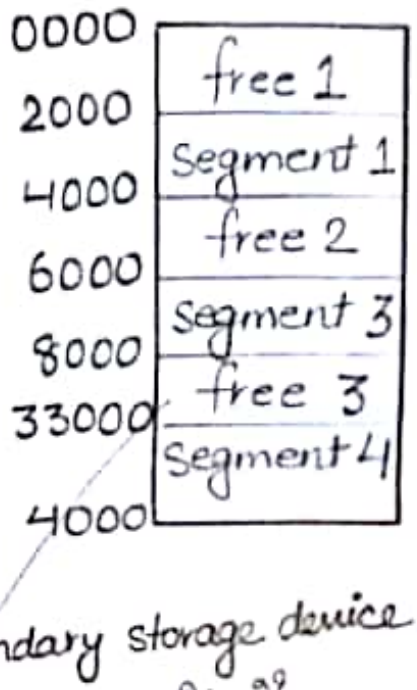
(6) Segmentation table is used for the validation of an address.

(7) A segmentation table has two field
   i) Base address of the segment.
   ii) Size of the segment.

(8) Base address of the segment is the base address of the location of a particular segment in the main memory from where segment stores.

(9) Segment Table

Segment no.	Base address	Size of segment
0	1000	800
1	6000	1000
2	3000	1200
3	4000	2000
4	2000	4000



0,2 secondary storage device में है।

According to this figure, various memory segments in the main memory and there corresponding entries in the segment table.

For Example- Segment 3 has loaded in the main memory.

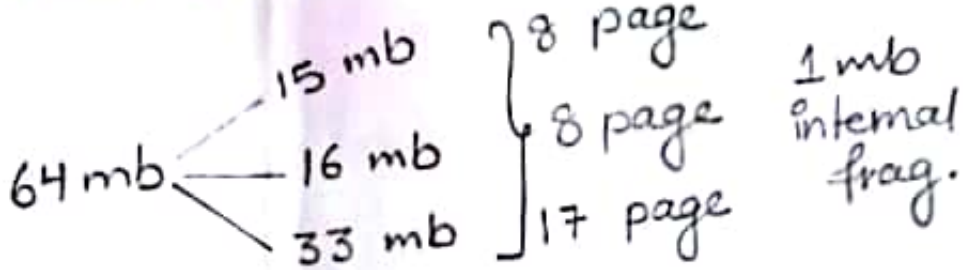
PAGING AND SEGMENTATION SYSTEM

(1) Segmentation of memory can be implemented with paging scheme.

(2) Unequal size of segment can be represented of equal size.

(3) For Example- In a logical address space of 64 mb, there are 32 page of 2 mb each. A program is divided into three segment of size 15mb, 16 mb, 33mb each.

These three segment require 8, 8, 17 pages where 1 mb is an internal fragmentation.



(4) A logical address has three part
i) segment no.
ii) page no.
iii) offset

Example-